

# Impossibility of secure cloud quantum computing for classical client

Tomoyuki Morimae<sup>1</sup> and Takeshi Koshiha<sup>2</sup>

<sup>1</sup>*ASRLD Unit, Gunma University, 1-5-1 Tenjin-cho Kiryu-shi Gunma-ken, 376-0052, Japan*

<sup>2</sup>*Graduate School of Science and Engineering, Saitama University,  
255 Shimo-Okubo, Sakura, Saitama 338-8570, Japan*

(Dated: July 8, 2014)

The first generation quantum computer will be implemented in the cloud style, since only few groups will be able to access such an expensive and high-maintenance machine. How the privacy of the client can be protected in such a cloud quantum computing? It was theoretically shown [A. Broadbent, J. F. Fitzsimons, and E. Kashefi, Proceedings of the 50th Annual IEEE Symposium on Foundation of Computer Science, 517 (2009)], and experimentally demonstrated [S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Science **335**, 303 (2012)] that a client who can generate randomly-rotated single qubit states can delegate her quantum computing to a remote quantum server without leaking any privacy. The generation of a single qubit state is not too much burden for the client, and therefore we can say that “almost classical client” can enjoy the secure cloud quantum computing. However, isn’t it possible to realize a secure cloud quantum computing for a client who is completely free from any quantum technology? Here we show that perfectly-secure cloud quantum computing is impossible for a completely classical client unless classical computing can simulate quantum computing, or a breakthrough is brought in classical cryptography.

## I. INTRODUCTION

Imagine that Alice who does not have any sophisticated quantum technology wants to factor a large integer. She has a rich friend, Bob, who has a full-fledged quantum computer. Alice asks Bob to perform her quantum computing on his quantum computer. However, the problem is that Bob is not a reliable person, and therefore she does not want to reveal her input (the large integer), output (a prime factor), and the program (Shor’s algorithm), to Bob. Can she delegate her quantum computing to Bob while keeping her privacy?

Recently, it was theoretically shown that such a secure cloud quantum computing is indeed possible [1]. (A proof-of-principle experiment was also demonstrated with photonic qubits [2].) In the protocol of Ref. [1] (Fig. 1), Alice, a client, has a device that emits randomly rotated single qubit states. She sends these states to Bob, the server, who has the full quantum technology. Alice and Bob are also connected with a classical channel. Bob performs quantum computing by using qubits sent from Alice, and classical messages exchanging with Alice via the classical channel. After finishing his quantum computation, Bob sends the output of his computation, which is a classical message, to Alice. This message encrypts the result of Alice’s quantum computing, which is not accessible to Bob. Alice decrypts the message, and obtains the desired result of her quantum computing. It was shown that whatever Bob does, he cannot learn anything about the input, the program, and the output of Alice’s computation [1, 3] (except for some unavoidable leakage, such as the upperbound of the input size, etc.).

In this protocol, the client has to possess a device that generates single qubit states. Generation of single qubit states is ubiquitous in today’s laboratories, and therefore not too much burden for the client. In other words,

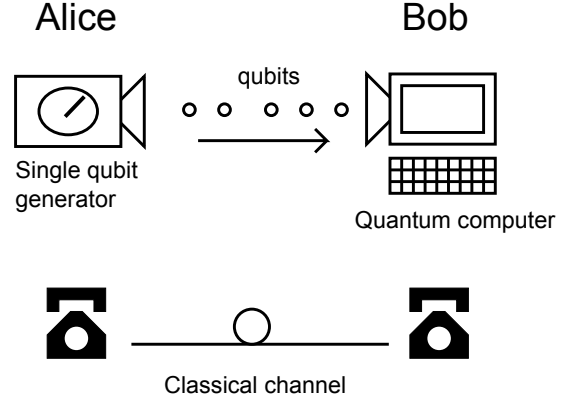


FIG. 1: The secure cloud quantum computing protocol proposed in Ref. [1]. Alice possesses a device that emits randomly-rotated single-qubit states. Bob has a universal quantum computer. Alice and Bob share a two-way classical channel.

“almost classical” client can enjoy secure cloud quantum computing.

However, isn’t it possible to realize secure cloud quantum computing for a completely classical client (Fig. 2)? Many variant protocols of secure cloud quantum computing have been proposed recently [4–14]. For example, it was shown that, in stead of single-qubit states, the client has only to generate weak coherent pulse states if we add more burden to the server [7]. Coherent states are considered as “more classical” than single-photon states, and therefore it enables secure cloud quantum computing for “more classical” client. It was also shown that secure cloud quantum computing is possible for a client who can only measure states [4] (Fig. 3). A measurement of a bulk state with a threshold detector is sometimes much

easier than the single-photon generation, and therefore the protocol also enables “more classical” client. However, these two protocols still require the client to have some minimum quantum technologies, namely the generation of weak coherent pulses or measurements of quantum states. In fact, all protocols proposed so far require the client to have some quantum ability, such as generation, measurement, or routing of quantum states [4–14]. (It is known that [1] if we have two quantum servers, a completely classical client can delegate her quantum computing. However, in this case, we have to assume that two servers cannot communicate with each other.) In other words, the possibility of the perfectly secure cloud quantum computing for a completely classical client has been an open problem.

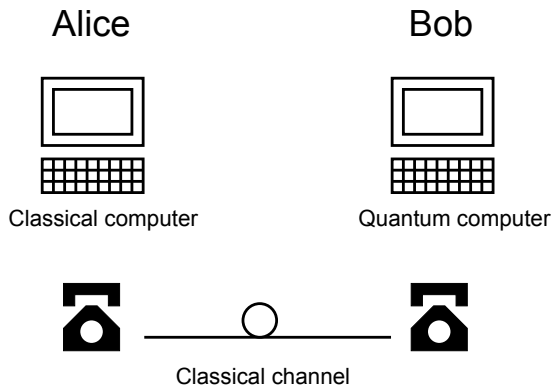


FIG. 2: The secure cloud quantum computing for a classical client. Alice has only a classical computer, whereas Bob has a universal quantum computer. Alice and Bob share a two-way classical channel.

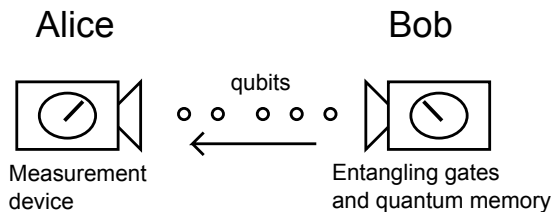


FIG. 3: The secure cloud quantum computing protocol proposed in Ref. [4]. Alice possesses a device that measure qubits. Bob has the ability of entangling operations and quantum memory.

In this paper, we show that perfectly-secure cloud quantum computing for a completely classical client is unlikely possible. Here, the perfect security means that an encrypted text gives no information about the plain text [16]. It is a typical security notion in the information theoretical security. The idea of the proof is as follows. Since no non-affine cryptography is known to be perfectly secure [16], we assume that the client uses an affine cryptography. We then show that if the cloud

quantum computing can be done in the perfectly secure way for a completely classical client, classical computing can efficiently simulate quantum computing. Although the conjecture of  $BPP \subsetneq BQP$  is not so solid as  $P \neq NP$  or that the polynomial hierarchy does not collapse, researchers in quantum computing believe  $BPP \subsetneq BQP$ . Therefore, we conclude that perfectly-secure cloud quantum computing is impossible for a completely classical client (unless a non-affine cryptography is shown to be perfectly secure or classical computing can efficiently simulate quantum computing).

## II. RESULT

Our setup is given in Fig. 2. Alice has only a classical computer (more precisely, the probabilistic polynomial time Turing machine), whereas Bob has a universal quantum computer. Furthermore, Alice and Bob share a two-way classical channel.

Let  $U$  be the  $n$ -qubit unitary operator that Alice wants to implement in her quantum computing, where  $n$  is a polynomial of the size of the input of her problem. (More precisely, she choses a unitary from the finite set  $\{U_j\}_{j=1}^r$  of unitaries, since the capacity of the classical channel between Alice and Bob is finite, and a set of finite unitaries is sufficient for universal quantum computing.) Without loss of generality, we can assume that the initial state of her quantum computing is the standard state  $|0\rangle^{\otimes n}$ . In other words, if she wants to start with a certain input state  $|\psi\rangle$ , the preparation of it is included in  $U$ . (In the secure cloud quantum computing protocol of Ref. [1], Alice can use unknown quantum state as the input, such as a given state from another person. However, in the present setup, by definition, Alice’s input is restricted to classical information. Therefore we assume that the input state is a standard one or she knows the classical description of the input quantum state.) Therefore, what Alice wants to hide from Bob are the classical description  $[U]$  of the unitary  $U$ , and the output of the computation, which is the computational basis measurement result on  $U|0\rangle^{\otimes n}$ . (The protocol of Ref. [1] allows Alice to finally obtain an output quantum state. However, again, we assume that Alice’s output is a classical information, since she is completely classical.)

In the setup of Fig. 2, what Alice and Bob can do is the following protocol.

1. Alice sends Bob the classical message

$$a = E([U], k)$$

that encrypts the classical description  $[U]$  of the unitary  $U$  with the private key  $k \in K$ , where  $K$  is the set of keys and  $E$  is an encrypting operation. Since the encryption is done by Alice, we assume that the key generation and the encrypting operation  $E$  can be done with a classical computer in  $\text{poly}(n)$  time. We also assume that the encryption operation  $E$  is an affine one.

- Bob performs quantum computing by using  $a$ , and obtains the output

$$b = Q(a)$$

of his quantum computation, which is a classical message. Here,  $Q$  is an operation that can be done with a quantum computer in  $\text{poly}(n)$  time. Bob sends  $b$  to Alice. The message  $b$  is an encrypted text of her quantum computing result, i.e., the measurement result on  $U|0\rangle^{\otimes n}$ .

- Alice decrypts  $b$  by calculating

$$c = D([U], k, b),$$

where  $D$  is the decrypting operation. Again, we assume that the decrypting operation  $D$  can be done with a classical computer in  $\text{poly}(n)$  time, since the decryption is done by Alice. The value  $c$  is the desired result of her quantum computing, i.e., the result of the measurement on  $U|0\rangle^{\otimes n}$ .

Now we show that if the above protocol can be done in the secure way, classical computing can efficiently simulate quantum computing by using a similar argument of Ref. [17]. Here, secure means the perfect security, i.e., Bob cannot gain any information from  $a$  about the classical description  $[U]$  of the unitary  $U$  (except for some necessarily leaking information such as the maximum size of  $[U]$ , etc.).

Let us assume that in the above protocol Alice wants to delegate a unitary  $U$ . Let us define

$$\xi \equiv E([I^{\otimes n}], k_0)$$

for a certain  $k_0 \in K$ , where  $I \equiv |0\rangle\langle 0| + |1\rangle\langle 1|$  is the two-dimensional identity operator. For any unitary  $U$ , there must exist a key  $k_U \in K$  such that

$$E([U], k_U) = \xi, \quad (1)$$

since otherwise Bob can learn that Alice's computation is not  $U$  when he receives  $\xi$  from Alice. This means that Bob can gain some information about Alice's unitary, which contradicts to the assumption of the perfect security [16].

Since we assume that  $E$  is an affine encryption, Alice can calculate  $k_U$  which satisfies Eq. (1) by herself for any  $U$  [18]. Furthermore, for the fixed value

$$Q(\xi) = Q(E([I^{\otimes n}], k_0)),$$

Alice can efficiently calculate

$$D([U], k_U, Q(\xi))$$

by herself by assumption. However, since

$$D([U], k_U, Q(\xi)) = D([U], k_U, Q(E([U], k_U))),$$

this means that Alice can efficiently calculate

$$D([U], k_U, Q(E([U], k_U)))$$

by herself, which is the result of the quantum computing  $U$ , i.e., the result of the measurement on  $U|0\rangle^{\otimes n}$ . This means that classical computing can efficiently simulate quantum computing. Therefore, if the above protocol can be done in the perfect secure way, classical computing can efficiently simulate quantum computing, which shows our claim.

### III. DISCUSSION

In this paper, we have shown that perfectly-secure cloud quantum computing is impossible for a completely classical client unless a non-affine cryptography is shown to be perfectly secure or classical computing can efficiently simulate quantum computing.

We think a majority of researchers in quantum information believe quantum computing is more powerful than classical computing. Furthermore, if classical computing could efficiently simulate quantum computing, a classical client can achieve perfectly secure cloud quantum computing in the following trivial way, and therefore our question, namely the possibility of the perfect-secure cloud quantum computing for a classical client, is trivialized.

1. The client encrypts a message which contains information about client's desired quantum computation, and sends it to the server via the classical channel.
2. The server returns the message to the client without doing anything.
3. The client decrypts the message, and does the quantum computing by herself.

At this moment, we do not know whether a perfectly secure non-affine cryptography is possible or not. Our result therefore does not exclude the possibility that in a future a perfectly-secure non-affine cryptography is found, and the non-affine cryptography enables perfectly-secure cloud quantum computing for a completely classical client in some clever way.

We also note that if we relax the requirement of the perfect security, there might be several possibilities of secure cloud quantum computing for a classical client. For example, we require not the perfect security but only the computation-theoretical security, fully-homomorphic encryption scheme [19] would be able to achieve secure cloud quantum computing.

### Acknowledgments

TM is supported by the Tenure Track System by MEXT Japan and KAKENHI 26730003 by JSPS. TK is

supported by KAKENHI 26540002, 24106008, 24240001, 23246071 by JSPS.

- 
- [1] A. Broadbent, J. F. Fitzsimons, and E. Kashefi, Universal blind quantum computation. Proc. of the 50th Annual IEEE Sympo. on Found. of Comput. Sci. 517 (2009).
  - [2] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Demonstration of blind quantum computing. Science **335**, 303 (2012).
  - [3] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner, Composable security of delegated quantum computation. arXiv:1301.3662
  - [4] T. Morimae and K. Fujii, Blind quantum computation for Alice who does only measurements. Phys. Rev. A **87**, 050301(R) (2013).
  - [5] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther, Experimental verification of quantum computation. Nature Phys. **9**, 727 (2013).
  - [6] J. F. Fitzsimons and E. Kashefi, Unconditionally verifiable blind computation. arXiv:1203.5217.
  - [7] V. Dunjko, E. Kashefi, and A. Leverrier, Blind quantum computing with weak coherent pulses. Phys. Rev. Lett. **108**, 200502 (2012).
  - [8] T. Morimae, V. Dunjko, and E. Kashefi, Ground state blind quantum computation on AKLT state. arXiv:1009.3486.
  - [9] T. Morimae and K. Fujii, Blind topological measurement-based quantum computation. Nature Communications **3**, 1036 (2012).
  - [10] T. Morimae, Continuous-variable blind quantum computation. Phys. Rev. Lett. **109**, 230502 (2012).
  - [11] V. Giovannetti, L. Maccone, T. Morimae, and T. G. Rudolph, Efficient universal blind computation. Phys. Rev. Lett. **111**, 230501 (2013).
  - [12] A. Mantri, C. Pérez-Delgado, and J. F. Fitzsimons, Optimal blind quantum computation. Phys. Rev. Lett. **111**, 230502 (2013).
  - [13] Q. Li, W. H. Chan, C. Wu, and Z. Wen, Triple-server blind quantum computation using entanglement swapping. Phys. Rev. A **89**, 040302(R) (2014).
  - [14] T. Sueki, T. Koshihara, and T. Morimae, Ancilla-driven universal blind quantum computation. Phys. Rev. A **87**, 060301(R) (2013).
  - [15] T. Morimae and K. Fujii, Secure entanglement distillation for double-server blind quantum computation. Phys. Rev. Lett. **111**, 020502 (2013).
  - [16] D. R. Stinson, *Cryptography: Theory and Practice*, (Chapman & Hall / CRC, 2006).
  - [17] V. Dunjko, T. Kapourniotis, and E. Kashefi, arXiv:1405.4558
  - [18] An affine transformation is a linear transformation plus a parallel translation. A candidate of the vector  $v(k_U)$  corresponding to a key  $k_U$  that satisfies  $E([U], k_U) = \xi$  is  $v(k_U) = v(\xi) - v([U])$ , where  $v([U])$  and  $v(\xi)$  are vectors corresponding to  $[U]$  and  $\xi$ , respectively. The vector (key)  $v(k_U)$  is a valid key, since a parallel translation is an invertible affine transformation. (An affine transformation is invertible if and only if the linear transformation is invertible. In this case, the linear transformation is the identity operation, and therefore invertible.)
  - [19] C. Gentry, Fully homomorphic encryption using ideal lattices, Symposium on the Theory of Computing (STOC) pp.169 (2009).